PTOCLM

11/15/04

AS

1. (original) A clock processing logic for determining an edge of a clock signal indicated in a sample vector by a bit location corresponding to a transition from one or more bits of a first value on one side of said bit location to one or more bits of a second value on another side of said bit location, wherein said bit location varies from cycle to cycle according to reference voltage and temperature variations affecting said clock signal, comprising:

edge detection logic configured to compare adjacent pairs of bits of said sample vector starting from one end of said sample vector to another end of said sample vector until a bit location corresponding to a transition from one or more bits of a first value on one side of said bit location to one or more bits of a second value on another side of said bit location is detected; and

sensitivity adjustment logic configured to adjust said bit location according to information of at least one other bit location corresponding to a previous cycle of said clock signal that was previously detected by said edge detection logic.

2. (original) The clock processing logic according to claim 1, wherein a prior bit location was previously detected by said edge detection logic during a prior cycle of said clock signal, and said sensitivity adjustment logic is configured to adjust said bit location according to the following logic:

if said bit location is less than said prior bit location by a predefined number of bit locations, then said bit location is adjusted to be said prior bit location decremented by one bit location; else

if said bit location is greater than said prior bit location by said predefined number of bit locations, then said bit location is adjusted to be said prior bit location incremented by one bit location; else

said bit location is adjusted to be said prior bit location.

3. (original) The clock processing logic according to claim 1, wherein prior bit locations have been previously detected by said edge detection logic during prior cycles of said clock signal, and said sensitivity adjustment logic is configured to adjust said bit location according to a moving average of a predefined number of said prior bit locations.

4. (original) The clock processing logic according to claim 1, further comprising a sensitivity counter incremented each cycle of said clock signal, wherein a prior bit location was previously detected by said edge detection logic during a prior cycle of said

clock signal, and said sensitivity adjustment logic is configured to adjust said bit location according to the following logic:

if said bit location is less than said prior bit location by a predefined number of bits and said prior bit location is greater than said predefined number of bits from a first bit location of said sample vector, then said bit location is adjusted to said prior bit location decremented by one bit location; else

if said bit location is greater than said prior bit location by said predefined number of bits and said prior bit location is greater than said predefined number of bits from a last bit location of said sample vector, then said bit location is adjusted to said prior bit location incremented by one bit location; else

if said sensitivity counter has not counted to a predefined sensitivity period count, then said bit location is adjusted to said prior bit location; else

if said sensitivity counter has counted to said predefined sensitivity period count, then said sensitivity counter is reset and if said bit location is less than said prior bit location and said prior bit location is not said first bit location of said sample vector, then said bit location is adjusted to said prior bit location decremented by one bit location; else

if said sensitivity counter has counted to said predefined sensitivity period count, then said sensitivity counter is reset and if said bit location is greater than said prior bit location and said prior bit location is not said last bit location of said sample vector, then said bit

location is adjusted to said prior bit location incremented by one bit location; else

said bit location is adjusted to said prior bit location.

5. (original) The clock processing logic according to claim 1, further comprising a metastability filter configured to correct errors in said sample vector due to metastability problems when capturing said sample vector, wherein said metastability filter processes said sample vector before providing said sample vector to said edge detection logic.

6. (original) The clock processing logic according to claim 5, wherein said metastability filter is configured to filter said sample vector for each bit location of said sample vector between high and low boundary bit locations so that the bit location being filtered is set to a value that is the predominant value of all bit locations between a low end filter bit location equal to the bit location less a predefined number of bit locations and a high end filter bit location equal to the bit location plus said predefined number of bit locations.

7. (original) The clock processing logic according to claim 6, wherein said high boundary bit location is equal to a highest bit location in said sample vector less said predefined number of bit locations, and said low boundary bit location is equal to a lowest bit

location of said sample vector plus said predefined number of bit locations.

8. (original) The clock processing logic according to claim 5, wherein said metastability filter is configured to filter said sample vector for each bit location of said sample vector between high and low boundary bit locations according to the following logic:

if a filter size is equal to three, and if a sum of values of the bit location being filtered and bit locations one less and one more than the bit location being filtered is greater than "1", then the bit location being filtered is set to "1"; else

if said filter size is equal to three, and if said sum of said values of the bit location being filtered and said bit locations one less and one more than the bit location being filtered is not greater than "1", then the bit location being filtered is set to "0".

9. (original) The clock processing logic according to claim 1, wherein said first value used in said edge detection logic is a "1" and said second value used in said edge detection logic is a "0".

10. (original) The clock processing logic according to claim 9, wherein said edge detection logic finds a first occurrence of said transition in said sample vector.

11. (original) The clock processing logic according to claim 1, further comprising an error adjustment logic configured to adjust said bit location detected by said edge detection logic by a number of bit locations as specified in a field of a control register readable by said error adjustment logic.

12. (original) The clock processing logic according to claim 1, further comprising a manual override logic configured to provide a user specified bit location indicating an edge of said clock signal when said clock processing logic is not enabled except for said manual override logic.

13. (original) The clock processing logic according to claim 1, wherein said edge detection logic and said sensitivity adjustment logic are implemented by hardwired logic circuits.

14. (original) The clock processing logic according to claim 1, wherein said edge detection logic and said sensitivity adjustment logic are implemented by a processing unit.

15. (original) The clock processing logic according to claim 1, wherein said edge detection logic and said sensitivity adjustment logic are implemented by a combination of hardwired logic circuits and a processing unit.

16. (original) A method for processing a sample vector indicating an edge of a clock signal by a bit location corresponding to a transition from one or more bits of a first value on one side of said bit location to one or more bits of a second value on another side of said bit location, wherein said bit location varies from cycle to cycle according to reference voltage and temperature variations affecting said clock signal, comprising:

detecting a bit location corresponding to a transition from one or more bits of a first value on one side of said bit location to one or more bits of a second value on another side of said bit location; and

adjusting said bit location according to information of at least one other bit location corresponding to a previous cycle of said clock signal that was previously detected.

17. (original) The method according to claim 16, wherein a prior bit location was previously detected during a prior cycle of said clock signal, and said adjusting said bit location comprises:

adjusting said bit location to be said prior bit location decremented by one bit location if said bit location is less than said prior bit location by a predefined number of bit locations; and

adjusting said bit location to be said prior bit location incremented by one bit location if said bit location is greater than said prior bit location by said predefined number of bit locations.

18. (original) The method according to claim 16, wherein prior bit locations have been previously detected during prior cycles of said clock signal, and said adjusting said bit location comprises adjusting said bit location according to a moving average of a predefined number of said prior bit locations.

19. (original) The method according to claim 16, wherein a prior bit location was previously detected by said edge detection logic during a prior cycle of said clock signal, and said adjusting said bit location comprises:

adjusting said bit location to said prior bit location decremented by one bit location if said bit location is less than said prior bit location by a predefined number of bits and said prior bit location is greater than said predefined number of bits from a first bit location of said sample vector;

adjusting said bit location to said prior bit location incremented by one bit location if said bit location is greater than said prior bit location by said predefined number of bits and said prior bit location is greater than said predefined number of bits from a last bit location of said sample vector;

adjusting said bit location to said prior bit location if a sensitivity counter incremented each cycle of said clock signal has not counted to a predefined sensitivity period count;

adjusting said bit location to said prior bit location decremented by one bit location and resetting said sensitivity counter if said sensitivity counter has counted to said predefined sensitivity period count, said bit location is less than said prior bit location and said prior bit location is not said first bit location of said sample vector; and

adjusting said bit location to said prior bit location incremented by one bit location and resetting said sensitivity counter if said sensitivity counter has counted to said predefined sensitivity period count, said bit location is greater than said prior bit location and said prior bit location is not said last bit location of said sample vector.

20. (original) The method according to claim 16, further comprising filtering said sample vector for each bit location of said sample vector between high and low boundary bit locations so that the bit location being filtered is set to a value that is the predominant value of all bit locations between a low end filter bit location equal to the bit location less a predefined number of bit locations and a high end filter bit location equal to the bit location plus said predefined number of bit locations.

21. (original) The method according to claim 20, wherein said high boundary bit location is equal to a highest bit location in said sample vector less said predefined number of bit locations, and said low boundary

bit location is equal to a lowest bit location of said sample vector plus said predefined number of bit locations.

22. (original) The method according to claim 16, further comprising filtering said sample vector for each bit location of said sample vector between high and low boundary bit locations by:

setting a bit location being filtered to "1" if a sum of values of the bit location being filtered and bit locations one less and one more than the bit location being filtered is greater than "1"; and

setting said bit location being filtered to a "0" if said sum of said values of the bit location being filtered and said bit locations one less and one more than the bit location being filtered is not greater than "1".

23. (original) The method according to claim 16, wherein said first value is a "1" and said second value is a "0".

24. (original) The method according to claim 23, wherein said detecting a bit location comprises finding a first occurrence of said transition in said sample vector.

25. (original) The method according to claim 16, further comprising: adjusting said bit location corresponding to said transition by a number of bit locations as specified in a field of a control register.

CLAIMS 26-46 (CANCELLED)